Session-based Recommendation with Dual Graph Networks

Tajuddeen Rabiu Gwadabe[1,*], Mohammed Ali Mohammed Al-hababi [1] and Ying Liu[1]

[1]School of Computer Science and Technology, University of Chinese Academy of Sciences (UCAS), China

Abstract

Session-based recommendation task aims at predicting the next item an anonymous user might click. Recently, graph neural networks have gained a lot of attention in this task. Existing models either construct a directed graph or a hypergraph and learn item embedding using some form of graph neural networks. We argue that constructing both directed and undirected graphs for each session may outperform either method since for some sessions the sequence of interaction may be relevant while for others it may not be relevant. In this paper, we propose a novel Session-based Recommendation model with Dual Graph Networks (SR-DGN). SR-DGN constructs a directed and an undirected graph from each session and learns both sequential and non-sequential item representation using sequential and non-sequential graph neural networks models respectively. Using shared learnable parameters, SR-DGN learns global and local user preferences for each network and uses the network with the best scores for recommendation. Experiments conducted on three real-world datasets showed its superiority over state-of-the-art models.

Keywords

session-based recommendation, graph neural networks, directed and undirected graphs,

1. Introduction

Recommender systems have become an essential component of the internet user experience as they assist consumers sift through the ever-increasing volume of information. Some online sites allow non-login users, however, the recommender systems have to rely on the current anonymous session exclusively for making recommendations. Session-based recommender systems aim at providing relevant recommendations to such anonymous users.

Recent developments in deep learning architectures have resulted in researchers focusing on using these architectures in session-based recommendation task. Recurrent neural networks [1] were first proposed to learn the sequential interaction between items in a session. More recently, Graph Neural Networks (GNN) have been proposed for session-based recommendation [2]. These models construct directed graphs for each session and learn item representation using the sequential Gated Graph Neural Networks (GGNN). On the other hand memory network models like STAMP [3] have shown that the order of the sequence may not be important in session-based recommendation and proposed session-based recommendation model that does not depend on the sequence of interactions. Similarly, hypergraph models like DHCN [4] proposed constructing a hypergraph for a session and learning item representation on a hypergraph convolutional network that also neglects the sequence of interactions between items.

This has led to two thought classes. Either consider the sequence of interactions between items since users interacted with items sequentially or neglect the sequence since item order may not be relevant since users interact with the items in an online setting. However, both thoughts have merit. For example, on an e-commerce site, buying a particular brand of phone might influence buying a screen guard - hence the sequence might be relevant. However, buying household supplies such as tissue might not influence buying any other particular item - hence the sequence might be irrelevant. We argue that the two thought classes might be complementary to each other. That is, for some sessions, considering the sequence is relevant while for some sessions it might be irrelevant.

To this end, we propose a Session-based Recommendation model with Dual Graph Networks, SR-DGN. SR-DGN first constructs two graph networks - a directed and an indirect graph for each session and learns item representation using sequential and non-sequential GNN models respectively. From the individual item representations, SR-DGN learns local and global user preferences. Each network will present a score for each item and the network with the best score is used for making the recommendation. Our main contributions are summarized as follows:

- SR-DGN proposed using two graph networks - directed and undirected graph for each session and learns item representations using sequential and non-sequential GNN models. For learn-

DL4SR'22: Workshop on Deep Learning for Search and Recommendation, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM), October 17-21, 2022, Atlanta, USA

*Corresponding author.

tgwadabe@mails.ucas.ac.cn (T. R. Gwadabe);
mohammed_al-hababi@mails.ucas.ac.cn (M. A. M. A.);
yingliu@ucas.ac.cn (Y. Liu)

ing sequential item representation, SR-DGN uses GGNN [5], while for learning non-sequential item representation, SR-DGN uses an SGC [6] with a gating highway connection.

- SR-DGN learns local and global user preferences from each graph network using shared learnable parameters between the networks. Then, each network in SR-DGN provides scores for each item and the network with the best score is used for making the recommendation.
- Experimental results on three benchmark datasets demonstrate the effectiveness of SR-DGN. Further analysis showed that while the sequential network performs better on some datasets, the non-sequential network perform better on others. These further prove the effectiveness of using both networks for the session-based recommendation task.

2. Related Works

Session-based recommendation models use the implicit temporal feedbacks of users such as clicks obtained by tracking user activities. Traditional machine learning models such as Markov Chain (MC) models have been used for sequential recommendation. Zimdars et al. [7] proposed extracting sequential patterns from sessions and predicting the next click using decision tree models. FMPC [8] generalizes MC method and matrix factorization to model short term user preference and long-term user preference respectively. However, MC models suffer from the assumption of an independence relationship between the states in a sequence and an unmanageable state space when considering all the possible sequences.

Recently, deep learning models have achieved the state-of-the-art performance in session-based recommendation. Hidasi et al. [1] first proposed GRU4Rec, a recurrent neural network for session-based recommendation. The model uses session-parallel mini-batches and pairwise ranking loss. Liu et al. [9] proposed NARM, which uses recurrent neural network with attention mechanism to learn both the local and the global user preference. Li et al. [3] proposed using memory networks and showed that modelling the sequential nature may not be necessary. Wu et al. [2] constructed directed graphs for each session and learned the local and global user preferences using GGNN. Wang et al. [10] constructs a hypergraph for each session and proposed a hypergraph attention network for recommendation.

3. SR-DGN

3.1. Problem Statement and Graph Construction

Session-based recommendation aims to predict the next click of an anonymous user session. For a dataset with distinct items set $V = v_1, v_2, \ldots, v_n$, let an anonymous session s be represented by the ordered list, $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,t-1}]$ where $v_{s,i} \in V$ is a clicked item within the session s, session-based recommendation aims to recommend the next item to be clicked, $v_{s,t}$. The output of SR-DGN is a ranked probability score for all the candidate items where the top-K items based on the probabilities $\hat{\mathbf{y}}$ will be recommended as the potential next clicks.

For each session s, our model constructs a directed and an undirected graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ and $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ respectively. For both graphs, $v_i \in V_s$ and $v_i \in V_n$ if v_i is clicked within the current session. A directed edge $(v_{i-1}, v_i) \in \mathcal{E}_s$ exists from v_{i-1} to v_i if item v_i was clicked immediately after v_{i-1}. An undirected edge $(v_{i-1}, v_i) \in \mathcal{E}_n$ exists between v_{i-1} to v_i if item v_i was clicked before or after item v_{i-1}. For the directed graph, we normalized the outgoing and incoming adjacency matrices by the degree of the outgoing node. The overview of the SR-DGN model is given in Figure 1.

3.2. Learning Sequential and Non-Sequential Item Embedding

We first transform all items $v_i \in V$ into a unified embedding space $v_i \in \mathbb{R}^d$, where d is the dimension size. Using this initial embedding, we learn sequential and non-sequential item embedding, v_{is} and v_{in} respectively.

3.2.1. Learning Sequential Item Embedding

We use GGNN [5] similar to SR-GNN [2] for learning the sequential item representation. Given the incoming and outgoing adjacency matrices and the initial item embedding, GGNN updates the item embedding as follows:

$$\mathbf{a}_{is}^t = \mathbf{A}_{is:}[\mathbf{v}_1^{t-1}, \ldots, \mathbf{v}_n^{t-1}]^T \mathbf{H}_1 + \mathbf{b}_1, \tag{1}$$

$$z_i^t = \sigma(\mathbf{W}_z \mathbf{a}_{is}^t \mathbf{U}_z \mathbf{v}_i^{t-1}), \tag{2}$$

$$r_i^t = \sigma(\mathbf{W}_r \mathbf{a}_{is}^t \mathbf{U}_r \mathbf{v}_i^{t-1}), \tag{3}$$

$$\hat{\mathbf{v}}_i^t = tanh(\mathbf{W}_o \mathbf{a}_{is}^t + \mathbf{U}_r(r_i^t \odot \mathbf{v}_i^{t-1})) \tag{4}$$

$$\mathbf{v}_{is} = (1 - z_i^t) \odot \mathbf{v}_i^{t-1} + z_i^t \odot \hat{\mathbf{v}}_i^t \tag{5}$$

where $\mathbf{A}_{is:} \in \mathbb{R}^{1x2n}$ is the i-th row of the incoming and outgoing matrices. $\mathbf{H}_1 \in \mathbb{R}^{dx2d}$ and $\mathbf{b}_1 \in \mathbb{R}^d$ are weight and bias parameters respectively. $z_i^t \in \mathbb{R}^{dxd}$ and $r_i^t \in \mathbb{R}^{dxd}$ are the reset and update gates respectively.

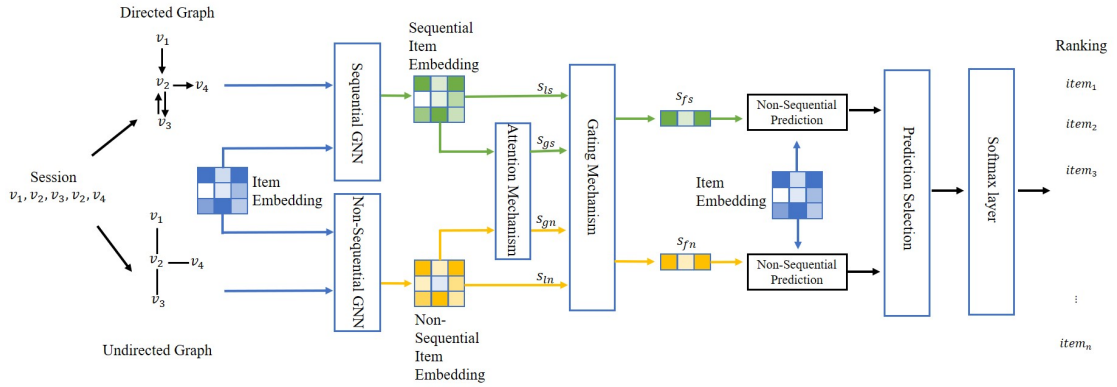

Figure 1: Overview of SR-DGN model. For each session s, directed and undirected graphs are constructed and sequential and non-sequential item representations are learned using sequential and non-sequential GNN models respectively. Using shared learnable parameters, the final sequential and non-sequential session representation is learned. Finally, the best prediction is selected for making recommendations.

3.2.2. Learning Non-Sequential Item Embedding

To learn the non-sequential item representation, v_{in} we used SGC [6] with a proposed highway connections. Formally, the update can is given by:

$$\mathbf{a}_{in}^{t} = \mathbf{A}_{in:}^{t}[\mathbf{v}_1^{t-1}, \ldots, \mathbf{v}_n^{t-1}]^T\mathbf{H}_2 + \mathbf{b}_2, \tag{6}$$

$$\mathbf{v}_{in} = \mathbf{g}_1 \odot \mathbf{a}_{in}^{t} + (1 - \mathbf{g}_1) \odot \mathbf{v}_i, \tag{7}$$

$$\mathbf{g}_1 = \mathbf{W}_{\mathbf{g}_1}([\mathbf{v}_{in}; \mathbf{v}_i]). \tag{8}$$

where $\mathbf{A}_{in:} \in \mathbb{R}^{1xn}$ is the i-th row of the undirected graph adjacency matrix. $\mathbf{H}_2 \in \mathbb{R}^{dxd}$ and $\mathbf{b}_2 \in \mathbb{R}^{d}$ are weight and bias parameters respectively. $\mathbf{g}_1$ is the gating mechanism used to improve the performance of the non-sequential item representation.

3.3. Learning Session Embedding

From the sequential and non-sequential item embedding, we learn the local and the global user preferences for each network using shared learnable parameters. Considering the sequential item embedding, to obtain the final session embedding, we use a gating mechanism that aggregates the global and the local user preferences. The sequential local user preference $\mathbf{s}_{ls}$ is obtained from the sequential embedding of the last clicked item while the sequential global preference $\mathbf{s}_{gs}$ is obtained from the sequential embedding of all clicked items in a session using additive an attention mechanism. Formally, the sequential global preference $\mathbf{s}_{gs}$ is given by;

$$\mathbf{s}_{gs} = \sum_{i}^{t} \alpha_i \mathbf{h}_{s,i}, \tag{9}$$

where $\mathbf{h}_{s,i}$ is the i-th sequential item embedding and α_i is the attention weight of the i-th timestamp given by:

$$\alpha_i = \mathbf{q}^T \sigma(\mathbf{W}_1\mathbf{h}_{s,t} + \mathbf{W}_1\mathbf{h}_{s,i} + b), \tag{10}$$

where the parameters $\mathbf{q}$,$\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable to control the additive attention. The final sequential session representation $\mathbf{s}_{fs}$ is obtained by aggregating the sequential local and global preferences using a gating mechanism. Formally, the final session representation $\mathbf{s}_{fs}$ is obtained as follows;

$$\mathbf{s}_{fs} = \mathbf{g}_2 \odot \mathbf{s}_{gs} + (1 - \mathbf{g}_2) \odot \mathbf{s}_{ls}, \tag{11}$$

$\mathbf{g}_2$ is the gating function obtained by;

$$\mathbf{g}_2 = \mathbf{W}_{\mathbf{g}_2}([\mathbf{s}_{gs}; \mathbf{s}_{ls}]) \tag{12}$$

$\mathbf{W}_{\mathbf{g}_2} \in \mathbb{R}^{2d \times d}$ is a trainable transformation matrix and [;] is a concatenation operation. From the non-sequential item embedding, using the same learnable parameters, the final non-sequential representation, $\mathbf{s}_{fs}$ can be obtained.

3.4. Making Recommendation

From the sequential and non-sequential final session representations, the sequential and non-sequential unnormalized scores of each candidate item $v_i \in V$ can be obtained by multiplying the item embedding $\mathbf{v}_i$ with the each the corresponding final session representation. The sequential unnormalized score $\hat{\mathbf{z}}_{is}$, is defined as:

$$\hat{\mathbf{z}}_{is} = \mathbf{s}_{fs}^T \mathbf{v}_i. \tag{13}$$

The non-sequential unnormalized score $\hat{\mathbf{z}}_{in}$ is obtained in similar way. For the recommendation, we use the sum of

the two unnormalized scores. A softmax is then applied to calculate the normalized probability output vector of the model $\hat{\mathbf{y}}_i$ as follows:

$$\hat{\mathbf{y}} = softmax(\hat{\mathbf{z}}) \tag{14}$$

where $\hat{\mathbf{z}} \in \mathbb{R}^n$ is the sum unnormalized score of the sequential and non-sequential scores and $\hat{\mathbf{y}} = \mathbb{R}^n$ is the probability of each item to be the next click in session s.

For any given session, the loss function is defined as the cross-entropy between the predicted click and the ground truth. The cross-entropy loss function is defined as follows:

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{n} \mathbf{y}_i log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) log(1 - \hat{\mathbf{y}}_i) \tag{15}$$

where $\mathbf{y}$ is the one-hot encoding of the ground truth items. Adam optimizer is then used to optimize the cross-entropy loss.

4. Performance Evaluation

In this section we aim to answer the following questions:

RQ1. How does the proposed SR-DGN model compare against the existing state-of-the-art baseline models?

RQ2. How does the proposed SR-DGN sequential and non-sequential networks compare against each other?

4.1. Experimental Configurations

4.1.1. Datasets

Three popular publicly available datasets, Yoochoose [1], RetailRocket[2] and Diginetica[3] were used to evaluate the performance of the proposed model. The Yoochoose dataset was obtained from the RecSys challenge 2015. RetailRocket dataset contains 6 months personalized transactions from an e-commerce site available on Kaggle while the Diginetica dataset is from the CIKM 2016 Cup. All datasets consist of transactional data from e-commerce sites. We used similar pre-processing with [2, 10] by removing the items occurring less than 5 times and the session of length less than 2. We used the last week transactions for testing in all datasets. Similar to existing models, we augment the training sessions by splitting the input sequence. For example, from the sequence $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$ we generate the following input sequence: $([v_{s,1}], v_{s,2}), \ldots, ([v_{s,1}, v_{s,2}, \ldots, v_{s,n-1}], v_{s,n})$ and used the most recent 1/64 portion of the Yoochoose dataset.

[1] http://2015.recsyschallenge.com/challege.html
[2] https://www.kaggle.com/retailrocket/ecommerce-dataset
[3] http://cikm2016.cs.iupui.edu/cikm-cup

4.1.2. Baseline

We compare the performance of our proposed SR-DGN model with traditional and deep learning representative baseline models. The traditional baseline model used is Factorized Personalized Markov Chain model (FPMC) [8]. The deep learning baselines include RNN-based models GRU4Rec [1], RNN with attention model (NARM) [9], memory-based with attention model (STAMP) [3], directed graph model SR-GNN [2] and hypergraph models DHCH [4] and SHARE [10]

4.1.3. Evaluation Metrics.

We used two common accuracy metrics, $P@K = 20, 10$ and $MRR@K = 20, 10$, for evaluation. P@K evaluates the proportion of correctly recommended unranked items, while MRR@K evaluates the position of the correctly recommended ranked items.

4.1.4. Hyperparameter Setup.

We used the same hyperparameters similar to previous models [2, 4, 10]. we set the hidden dimension in all experiments to $d = 100$, learning rate for Adam optimizer set to 0.001 with a decay of 0.1 after every 3 training epochs. l_2 norm and batch size were set to 10^{-5} and 100 respectively on all datasets.

4.2. Comparison with Baseline

We compare the performance of SR-DGN with the existing baseline models in terms of $P@K = 20, 10$ and $MRR@K = 20, 10$ on Yoochoose 1/64, RetailRocket and Diginetica datasets. Table 1 shows the performance with the best performance highlighted in boldface. It can be seen that SR-DGN outperforms the best baseline models on all datasets. It is evident that, using both directed and undirected graphs can potentially improve the overall performance of graph neural network models for session-based recommendation.

From Table 1, it can also be seen that all deep learning models outperformed FPMC the traditional model except GRU4Rec. It can also be seen that on the RetailRocket dataset, STAMP (non-sequential model) outperformed NARM (sequential model). However, on the Diginetica dataset, the reverse case can be observed. These performances support our argument that both the sequential and non-sequential architecture for learning item representation can be complementary. Despite the simple architecture of our sequential and non-sequential models, SR-DGN was able to outperform more complex models like DHCN that uses self-supervised learning with both intra- and inter- session information.

Table 1
Performance of SR-DGN compared with other baseline models. The boldface is the best result over all methods and * denotes the significant difference for t-test. (all values are in percentages)

Dataset	Metrics	Models									
		FMPC	GRU4Rec	NARM	STAMP	SR-GNN	S^2-DHCN	SHARE	GGNN	SGC	SR-DGN
Yoochoose 1/64	P@20	45.62	60.64	68.32	68.74	70.57	70.39	71.17	71.06	71.32	**71.70**
	MRR@20	15.01	22.89	28.63	29.67	30.94	29.92	31.06	31.32	31.27	**31.51**
	P@10	32.01	52.45	57.50	58.07	60.09	59.18	60.59	60.60	60.71	**61.29***
	MRR@10	14.35	21.53	27.97	28.92	30.69	28.54	30.78	30.69	30.52	**30.79**
Diginetica	P@20	26.53	29.45	49.70	45.64	50.73	53.18	52.73	51.83	52.68	**53.42***
	MRR@20	6.95	8.33	16.17	14.32	17.59	18.44	18.05	17.99	18.63	**18.66**
	P@10	15.43	17.93	35.44	33.98	36.86	39.87	39.52	38.62	39.65	**40.20***
	MRR@10	6.20	7.33	15.13	14.26	15.52	17.53	17.12	17.07	17.73	**17.75**
RetailRocket	P@20	32.37	44.01	50.22	50.96	50.32	53.66	54.00	54.55	54.72	**55.85***
	MRR@20	13.82	23.67	24.59	25.17	26.57	27.30	27.12	29.39	29.23	**29.77**
	P@10	25.99	38.35	42.07	42.95	43.21	46.15	46.21	47.36	47.44	**48.20***
	MRR@10	13.38	23.27	24.88	24.61	26.07	26.85	26.61	28.98	28.73	**29.24***

4.3. Comparison with GGNN and SGC

We compare the performance of GGNN [5] (sequential) and SGC [6] (non-sequential) models with SR-DGN on all the three datasets in terms of $P@K = 20, 10$ and $MRR@K = 20, 10$. Table 1 shows that on all the datasets, the combined SR-DGN model outperformed both GGNN and SGC. Generally SGC outperforms GGNN on Precision metrics while GGNN outperforms SGC on MRR metrics. To ensure good performance of different datasets, considering both the sequential and non-sequential models as in the case of our proposed SR-DGN may be the solution.

4.4. Ablation Study

SR-DGN uses summation to aggregate the sequential and non-sequential unnormalized scores. We compare the performance of summation to max aggregation method. Table 2 shows the performance of summation aggregation method against max aggregation. It can be seen that on all datasets and on metrics, the summation method outperforms the max methods. It results are intuitive since with summation, items with overall highest probabilites are recommended. It also further demonstrate the advantage of using both the sequential and non-sequential networks in SR-DGN.

Table 2
Performance comparison of aggregation methods in SR-DGN (all values are in percentages)

Datasets	Metrics	Max	Sum
Yoochoose 1/64	P@20	71.65	**71.70**
	MRR@20	31.19	**31.51**
	P@10	61.16	**61.29**
	MRR@10	30.45	**30.79**
Diginetica	P@20	52.73	**53.42**
	MRR@20	18.45	**18.66**
	P@10	39.69	**40.20**
	MRR@10	17.54	**17.75**
RetailRocket	P@20	55.11	**55.85**
	MRR@20	29.55	**29.77**
	P@10	49.01	**48.20**
	MRR@10	29.06	**29.24**

5. Conclusion

In this paper, we proposed SR-DGN, a graph neural network model for session-based recommendation. SR-DGN constructs a directed and an undirected graph for each session and learns sequential and non-sequential item embedding using sequential and non-sequential GNN respectively. Using shared learnable parameters, SR-DGN learns the global and local user preferences from each of the item embedding learnt. For making recommendation, SR-DGN selects the max of the sequential and non-sequential scores. Experimental results showed that, SR-DGN outperformed state-of-the-art models on three benchmark datasets. Further analysis revealed that, for some datasets, non-sequential model outperforms sequential and the reverse is true for some other datasets. SR-SGN takes advantage of both scenarios to achieve better performance.

Acknowledgments

This project was partially supported by Grants from Natural Science Foundation of China 62176247. It was also

supported by the Fundamental Research Funds for the Central Universities and CAS/TWAS Presidential Fellowship for International Doctoral Students.

References

[1] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, 2016. arXiv:1511.06939.

[2] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, T. Tan, Session-based recommendation with graph neural networks, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 346–353. URL: http://dx.doi.org/10.1609/aaai.v33i01.3301346. doi:10.1609/aaai.v33i01.3301346.

[3] Q. Liu, Y. Zeng, R. Mokhosi, H. Zhang, Stamp: Short-term attention/memory priority model for session-based recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1831–1839. URL: https://doi.org/10.1145/3219819.3219950. doi:10.1145/3219819.3219950.

[4] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, X. Zhang, Self-supervised hypergraph convolutional networks for session-based recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI '21, 2021, pp. 4503–4511. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16578.

[5] Y. Li, R. Zemel, M. Brockschmidt, D. Tarlow, Gated graph sequence neural networks, in: Proceedings of ICLR'16, 2016.

[6] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph convolutional networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 6861–6871. URL: http://proceedings.mlr.press/v97/wu19e.html.

[7] A. Zimdars, D. M. Chickering, C. Meek, Using temporal data for making recommendations, in: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, UAI'01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, p. 580–588.

[8] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in: Proceedings of the 19th International Conference on World Wide Web, WWW '10, Association for Computing Machinery, New York, NY, USA, 2010, p. 811–820. URL: https://doi.org/10.1145/1772690.1772773. doi:10.1145/1772690.1772773.

[9] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 1419–1428. URL: https://doi.org/10.1145/3132847.3132926. doi:10.1145/3132847.3132926.

[10] J. Wang, K. Ding, Z. Zhu, J. Caverlee, Session-based recommendation with hypergraph attention networks, Proceedings of the 2021 SIAM International Conference on Data Mining (SDM) (2021) 82–90. URL: http://dx.doi.org/10.1137/1.9781611976700.10. doi:10.1137/1.9781611976700.10.